Rejections Under 35 U.S.C. § 102(e)

Claims 1-4, 11-18, 20, 27-35, 37, 38, and 40 of Patent Application Number 09/820,185 ("Application") were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,715,145 to Bowman-Amuah ("Bowman"). A proper rejection of a claim under 35 U.S.C. § 102 requires that a single prior art reference disclose each element of the claim. See, e.g., W.L. Gore & Assoc., Inc. v. Garlock, Inc., 721 F.2d 1540, 220 USPQ 303, 313 (Fed. Cir. 1983). Anticipation requires that every element of the claimed invention be disclosed in a single prior art reference. See, e.g., In re Paulsen, 30 F.3d 1475, 31 USPQ2d 1671 (Fed. Cir. 1994); In re Spada, 911 F.2d 705, 15 USPQ2d 1655 (Fed. Cir. 1990). For anticipation, there must be no difference between the claimed invention and the prior art reference disclosure as viewed by a person of ordinary skill in the field of the invention. See, e.g., Scripps Clinic & Res. Found. v. Genentech, Inc., 927 F.2d 1565, 18 USPQ2d 1001 (Fed. Cir. 1991).

The independent claims of the Application, claims 1, 18, 34, and 38, each contain the limitations of an object model comprising components, one or more metaprograms reflecting a computer system architecture, and a meta-machine binding the components to the metaprograms to generate a software system. Therefore, Applicants' remarks with respect to claim 1 shall also apply to each independent claim of the Application.

Independent Claims 1, 18, 34, and 38

The Examiner rejected claim 1 as anticipated by Bowman and relied on several points of rejection. Therefore, each point of rejection is discussed in turn.

Claim 1 of the application is drawn to a novel and non-obvious software generation system known as a meta-development environment (MDE) which allows a user to develop software systems by combining an object model with one or more metaprograms. Application, page 5, line 18 to page 6, line 20. Prior art object modeling code generation systems used code generators which were tied to ***specific architectures.*** Claim 1 distinguishes over prior art object modeling systems by ***abstracting out*** target ***system architectures*** into one or more ***metaprograms***, allowing the MDE to take an object model and generate computer code corresponding to the system architecture represented by a supplied metaprogram. Application, page 5, line 18 to page 6, line 20.

Claim 1 of the Application recites in relevant part:

> A method for developing a software system, comprising the steps of:
> . . .
> providing a set of one or more ***metaprograms*** reflecting a computer system ***architecture***; (Emphasis added).

The term ***architecture*** as in claim 1 of the Application refers to the unique ***combination*** of hardware and software in a particular computer system. Application, page 10, lines 12-27. For example, two systems having identical hardware and operating systems but different database software constitute two different ***architectures*** according to the Application. Application, page 10, lines 19-22. Thus, a metaprogram ***separates*** architecture from application, allowing the MDE to be an ***architecture independent*** software generation system. Application, page 9, lines 21-22.

A ***metaprogram*** as in claim 1 of the Application is an object used to reflect or represent a computer system ***architecture***. In other words, a metaprogram is an abstraction which represents a complete system architecture. In one embodiment, a metaprogram is described in the Application as a mixture of traditional programming language code and metacode, allowing the invention of the Application to reflect a computer system architecture. The Application discusses one embodiment of a metaprogram as:

> A metaprogram is a mix of metacode expressed in a traditional programming language and output (code). In a metaprogram, metacode is distinguishable from the code by indicating the beginning of the metacode with an opening delimiter (e.g., the characters "<%") and indicating the end of the metacode by a closing delimiter (e.g., the characters "%>"). For example, an example of a metaprogram 14 written in JAVA and outputting hypertext mark-up language (HTML) is shown below:
>
> ```
> <%
> If (attribute.hasSterotype("text") {
> %>'
>     <input type="text" name="<%=attribute.name%>"...
> <%
> //more metacode here
> %>
> ```
>
> Application, page 13, line 23 to page 14, line 8.

The Examiner rejects claim 1 as anticipated by Bowman by asserting that Bowman discloses metaprograms as in claim 1 of the Application.

With regard to claim 1, the Examiner states:

> As per claim 1, Bowman discloses a method for developing a software system, comprising the steps of:
> . . .
> - providing a set of one or more **metaprograms** reflecting a computer system architecture (col. 176 line 47, "the **meta-model** (i.e. **metaprograms**)", **reflects a computer system architecture**).
> (Emphasis added).

To support a rejection of claim 1, the Examiner asserts that Bowman discloses at col. 176, line 47, ***metaprograms*** as in the Application by citing the language "the meta-model (i.e. ***metaprograms***)." See Office Action, page 3.

The language from Bowman cited by the Examiner, including additional lines for context, states in relevant part:

> The present invention uses a robust process for developing component-based solutions. It includes deliverables that are above and beyond the Unified Modeling Language (UML). Furthermore, projects often customize it. The environment must provide the ability to extend the information model (i.e., the meta-model). Bowman, 176:41-47.

The Applicants assert that the cited language from Bowman ***nowhere*** contains or discloses the term ***metaprogram*** as asserted by the Examiner. Applicants also respectfully assert that Bowman nowhere discloses ***metaprograms*** which reflect a computer system ***architecture*** by combining a traditional programming language with metacode as in the Application. Thus, Applicants respectfully request clarification from the Examiner regarding this citation, and assert that Bowman nowhere teaches or discloses a ***metaprogram*** reflecting a computer system ***architecture*** as in claim 1 of the Application.

To support a rejection of claim 1 as anticipated by Bowman, the Examiner asserts that Bowman discloses a ***meta-machine*** which generates code by binding components to a metaprogram as in the Application.

With regard to claim 1, the Examiner states:

> As per claim 1, Bowman discloses a method for developing a software system, comprising the steps of:
> . . .
> - a ***meta-machine*** binding the components to the ***metaprograms*** to generate the software system for a computer system having said architecture (col. 177 lines 3-7, "Code generation. The ability to generate the application . . . (**by binding the components to the meta programs such that**) . . . a change to the model is a change to the code"). (Emphasis added).

The lines cited from Bowman actually state:

> Code generation. The ability to generate the application structure from the model is essential to high productivity. Furthermore, this step should be transparent to the user. As far as the use is concerned, a change to the model is a change to the code. Bowman, 177:3-7.

Applicants respectfully assert that the cited language from Bowman ***nowhere*** discloses a ***meta-machine*** as in claim 1 of the Application. Applicants further assert that Bowman nowhere discloses a ***meta-machine*** binding ***components*** to ***metaprograms*** as in the Application. Thus, Applicants respectfully request clarification from the Examiner regarding the above citation, and assert that Bowman nowhere teaches or discloses a ***meta-machine*** nor a ***meta-machine*** binding ***components*** to ***metaprograms*** as in claim 1 of the Application.

In light of the above arguments, Applicants respectfully assert that independent claims 1, 18, 34, and 38 are allowable for at least that reasons that Bowman does not teach or disclose ***metaprograms***, ***meta-machines***, or a ***meta-machine*** binding ***components*** to ***meta-programs*** as in claim 1 of the Application.

Dependent Claims 2-4, 11-17, 20, 27-33, 35, 37, 38, and 40

Dependent Claims 2-4, 11-17, 20, 27-33, 35, 37, 38, and 40 were also rejected under § 102(e) in view of Bowman, but are allowable for at least the reason that they each depend directly or indirectly from allowable independent claim 1, 18, 34, or 38.

Rejections Under 35 U.S.C. § 103

Claims 5-10, 19, 21-26, 36, and 39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Bowman in view of Mueller, "Instant UML", Wrox Press, 1997 ("Mueller"). For a claim to be properly rejected under 35 U.S.C. § 103, "[t]he PTO has the burden under section 103 to establish a prima facie case of obviousness. It can satisfy this burden only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references." *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988) (Citations omitted). Further, "[t]he mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art suggested the desirability of the modification." *In re Fritch*, 972 F.2d 1260, 1266, 23 U.S.P.Q.2d 1780 (Fed Cir. 1992).

Applicants respectfully assert that dependent claims 5-10, 19, 21-26, 36, and 39 are allowable over Bowman in view of Mueller for at least the reason that Bowman and Mueller do not teach or disclose each limitation of these claims. Specifically, claims 5-10, 19, 21-26, 36, and 39 are allowable for at least the reason that Bowman and Mueller do not teach or disclose ***metaprograms***, ***meta-machines***, or a ***meta-machine*** binding ***components*** to ***meta-programs***.

Applicants respectfully submit that claims 5-10, 19, 21-26, 36, and 39 are allowable over Bowman in view of Mueller for at least the reason that the Office Action fails to establish a prima facie case of obviousness, given that the cited references fail to teach or disclose every limitation of the rejected claims.

CONCLUSION

For at least the foregoing reasons, Applicants respectfully request that all outstanding rejections be withdrawn and that all pending claims of this application be allowed to issue. If the Examiner has any comments regarding Applicants' response or intends to dispose of this matter in a manner other than a notice of allowance, Applicants request that the Examiner telephone Applicants' undersigned attorney.

Respectfully submitted,

NEEDLE & ROSENBERG, P.C.

Gregory J. Kirsch
Registration No. 35,572

NEEDLE & ROSENBERG, P.C.
Customer Number 23859
(678) 420-9300
(678) 420-9301 (fax)

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Fee Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

Gregory J. Kirsch — 8 JUNE 2005 — Date